CCS-1 ASC Overview

Rich Graham

Group Leader (Acting)

Los Alamos
NATIONAL LABORATORY

CCS-1 Teams

Rich Graham, Group Leader (acting)
Jean Knowlton & Debra Goldstein, Administrative Specialists

CLUSTER RESEARCH

Ron Minnich, Team Leader
Sean Blanchard
Latchesar Ionkov
Li-Ta Lo (Ollie)
Andrey Mirtchovski
Craig Rasmussen
Chris Rickett
Sushant Sharma
Matthew Sottile
Greg Watson

Jason Mastaler, OSCR

Cindy Sievers, Access Grid

APPLICATION COMMUNICATIONS PERFORMANCE RESEARCH

David Daniel, Team Leader
Sami Ayyorgun
Brian Barrett
James Barker
Pallab Datta
Josh Hursey
Galen Shipman

APPLICATION SPECIFIC ARCHITECTURES

Maya Gokhale, Team Leader
Zack Baker
Chung-Hsing Hsu
Kris Peterson
Justin Tripp
Christof Teuscher
Brian van Essen
Keith Mosher

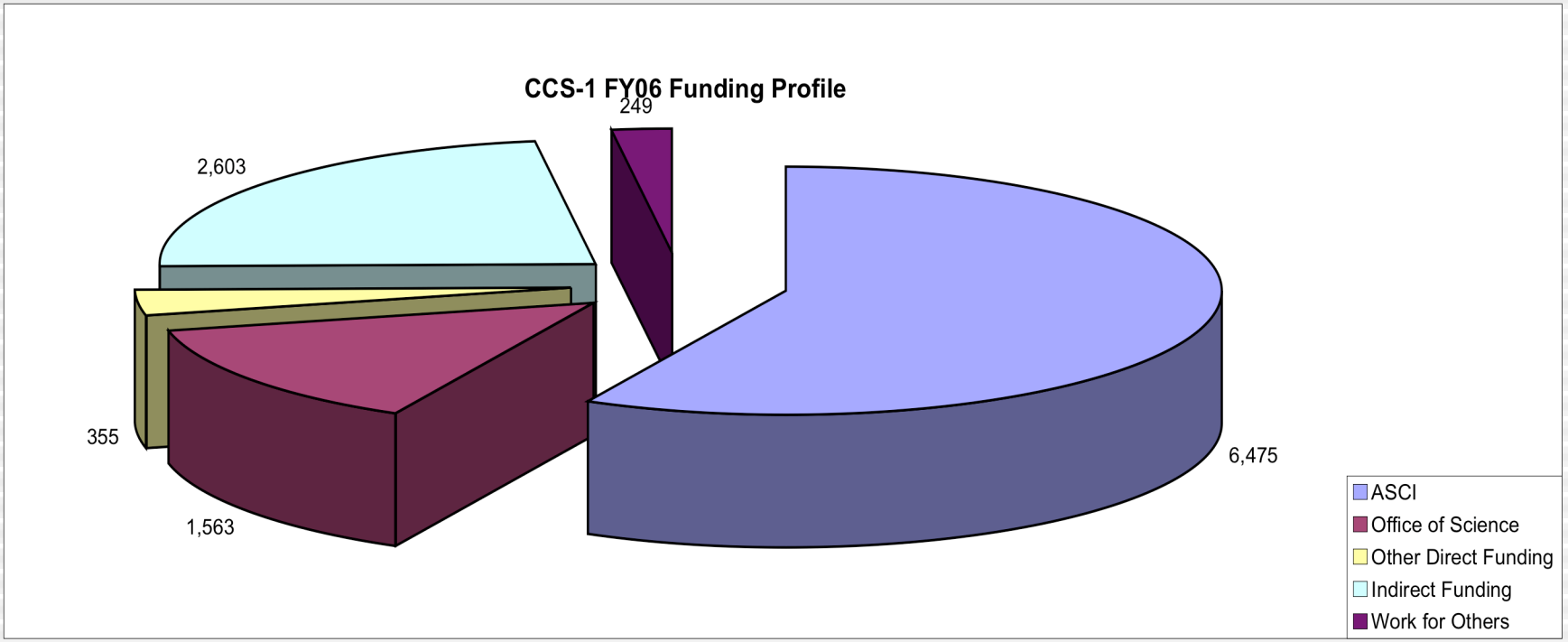
VISUALIZATION

Jim Ahrens, Team Leader
Nehal Desai
Jeff Inman (CCN-12)
Pat McCormick
Allen McPherson
John Patchett
Richard Strelitz

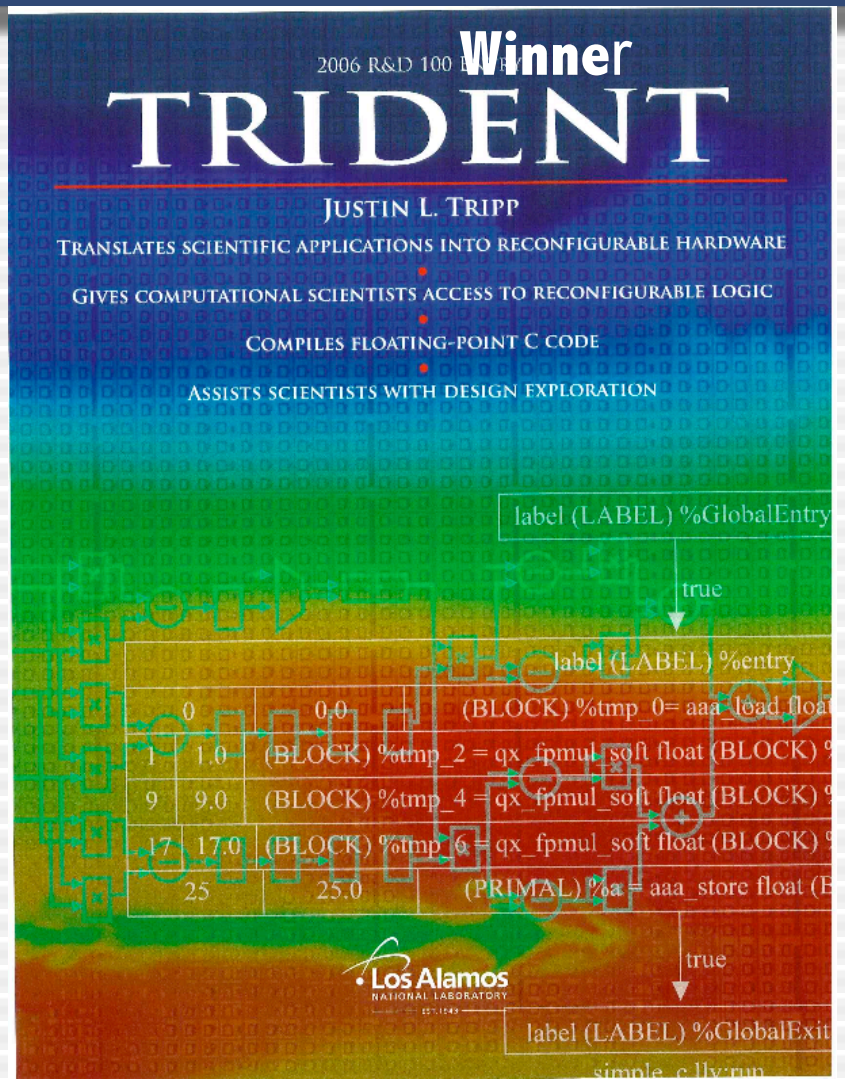
Group Demographics

- By Job Classification
  - TSM - 24
  - ASM - 2
  - Tech - 2
  - PD - 3
  - GRA - 4
  - UGS - 2
- By Degree (Technical)
  - PhD - 17
  - Masters - 10
  - BS/BA - 2

Funding Profile



2006 R&D 100 Winner



Group Focus

- Infrastructure
  - ♣Systems - automation
  - ♣Applications
- Performance
- New Architectures

Cluster Research

Team Focus

- System Software
  - ♣Science Appliance
- Mobile networks
- Tools
  - ♣Eclipse Parallel Tools Platform
  - ♣Language “tools”

System Software

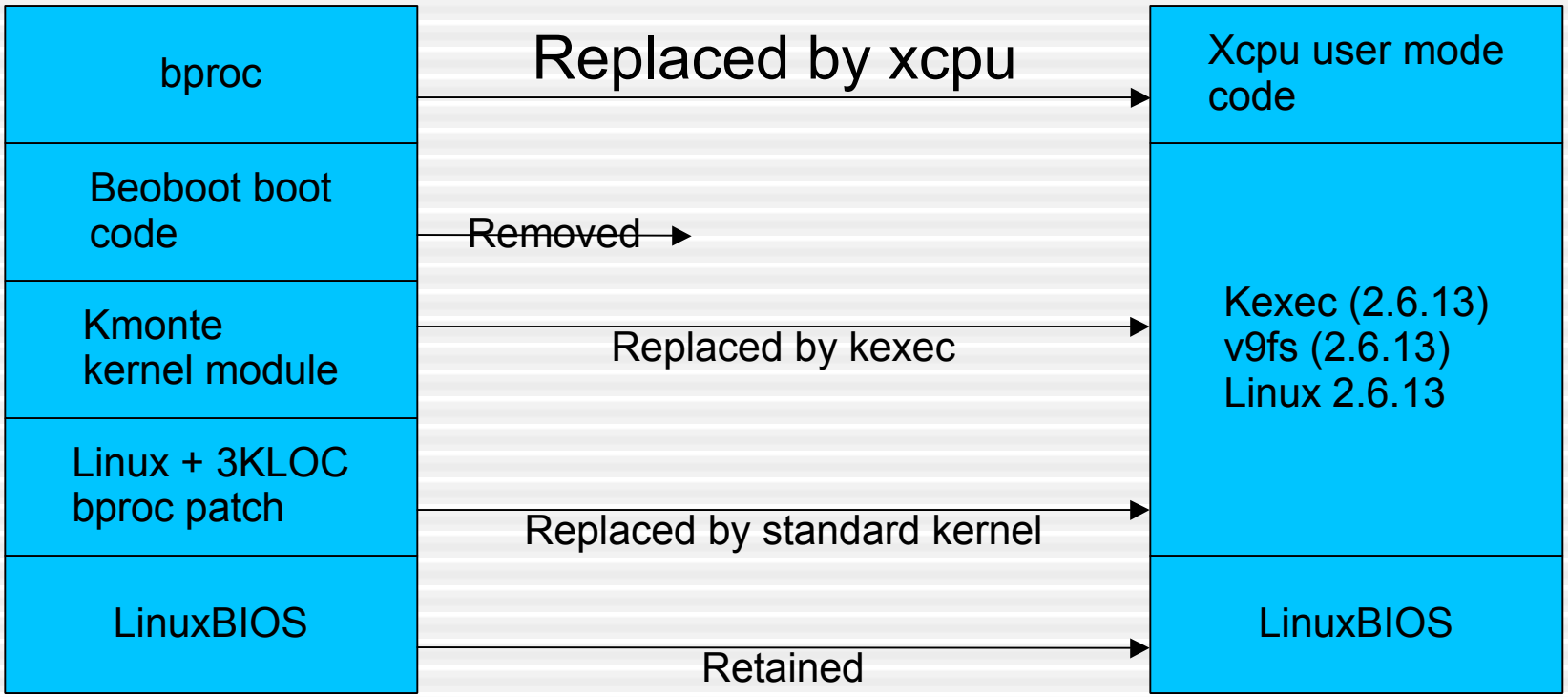
Team Strengths

- Many years of experience in O/S R&D (up to 30 years/person)
- Experience with different O/S's
- Part of the Linux "inner circle"
- Products deployed into production
- Training HPC-7/9 in Linux internals

Science Appliance

- Motivation: Reduce system management costs by an order of magnitude, and increase scalability of system management tools

Science Appliance Stack Today and "Today"



Actively being considered to run a couple Peta-Scale systems

BProc job startup time

- 16 Mbyte binary
- Measured performance, at LLNL, on SLURM: 125 seconds/400 nodes
- SLURM modified with bproc-style push: 22 secs
- Bproc: 3 seconds, 1024 nodes

Supermon - System Monitoring

- System monitoring s/w
- Efficient
- low-impact on system
- Extensible interface
- TAU and Supermon integration for correlating system and application level data
- Data analysis interface to Ganglia, and the TAU ParaProf tool

Mobile networks

- Plan9 based distributed sensor network
- Collaboration with ISR
- Uses cell phone modem and network for communications
- Plan9 provides secure distributed computing foundation

Tools

Eclipse Foundation History

- Originally developed by Object Technology International (OTI) and purchased by IBM for use by internal developers
- Released to open-source community in 2001, managed by consortium
  - Eclipse Public License (EPL)
  - Based on IBM Common Public License (CPL)
- Consortium reorganized into independent not-for-profit corporation, the Eclipse Foundation, in early 2004
  - Participants from over 85 companies

Eclipse Foundation Strategic Members

- Actuate Corporation
- BEA
- Borland
- Computer Associates
- Hewlett Packard
- IBM
- Intel
- MontaVista Software
- SAP AG
- Scapa Technologies
- Serena Software
- Sybase, Inc.
- Wind River

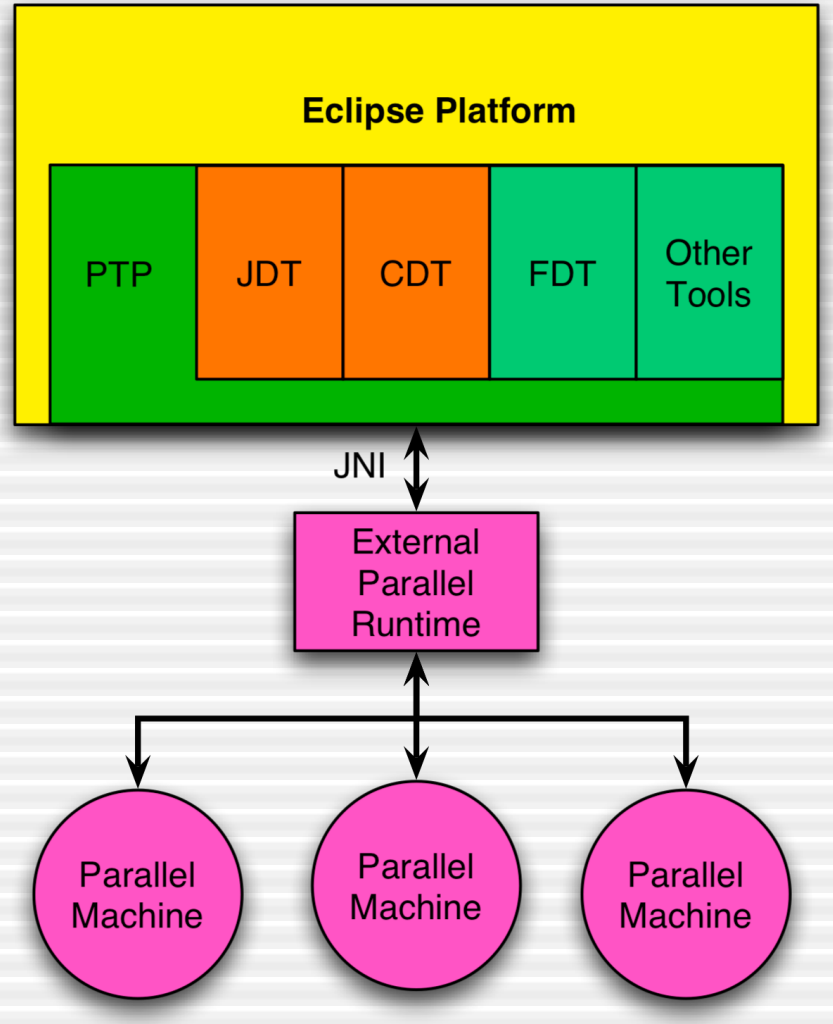
Parallel Tools Platform Objectives

- Extend Eclipse to support parallel development tools
- Equip Eclipse with key tools needed to start developing parallel codes
- Encourage existing parallel tool projects to support Eclipse
- Exploit enhanced capabilities to develop a new generation of parallel tools

Parallel Tools Platform Components

- Parallel Execution Environment
  - Extends existing execution environment to support parallel programs
- Parallel Debugger
  - Adds parallel debugging support to Eclipse
- Tools Integration
  - Support the integration of a variety of parallel tools, e.g. performance, verification, visualization, components
- Fortran
  - Adds Fortran support to a similar level as C/C++

Parallel Tools Platform Architecture



“Languages”

- Co-Array-Fortran
  - Goal: Provide standards based alternative for applications communication
  - Member of the Fortran standards committee

“Languages” - Cont’d

- Chasm - Language transformation system
  - Basis for the Fortran 2003 Fortran/C interoperability standard

- Used to:
  - Create 200 K lines of code for the Open MPI Fortran 90 MPI bindings and unit tests
  - Generate proposed new Fortran 2003 MPI bindings
  - Automatically create CCA components from legacy code
  - Generate Fortran wrappers for the C++ Visualization ToolKit
  - Create a library enabling Numeric Python users to seamlessly interoperate with Fortran arrays.

Application Communications and Performance Research

Team Focus

- Applications Communictions
  - LA-MPI
  - Open MPI
- Run-time Systems
  - Open Run-Time-Environment (RTE)

LA-MPI

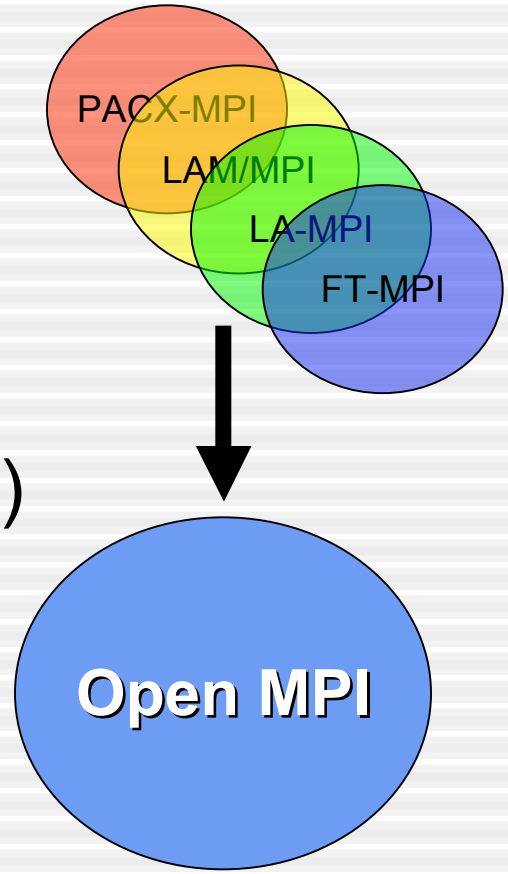
- Purpose: Deal with system failures the vendor was not addressing (1999)
- Full MPI 1.2 standard
- In use on most of LANL's Linux simulation platforms
- Some commercial interest (we declined, with Open MPI on the horizon)
- Features:
  - Network data integrity (high performance - much lower overhead than TCP/IP)
  - Data striping
  - Thread safe
- Shortcomings
  - S/W design for extensibility (first generation MPI)

MPI From Scratch!

- Developers of FT-MPI, LA-MPI, LAM/MPI
  - Kept meeting at conferences in 2003
  - Culminated at SC 2003: Let's start over
  - Open MPI was born
- Started serious design and coding work January 2004
  - All of MPI-2 (initially skipped one-sided ops)
  - Demonstrated at SC 2004
  - Released at SC 2005

MPI From Scratch: Why?

- Merger of ideas from
  - FT-MPI (U. of Tennessee)
  - LA-MPI (Los Alamos)
  - LAM/MPI (Indiana U.)
  - PACX-MPI (HLRS, U. Stuttgart)



Open MPI Collaborators

Operating Systems

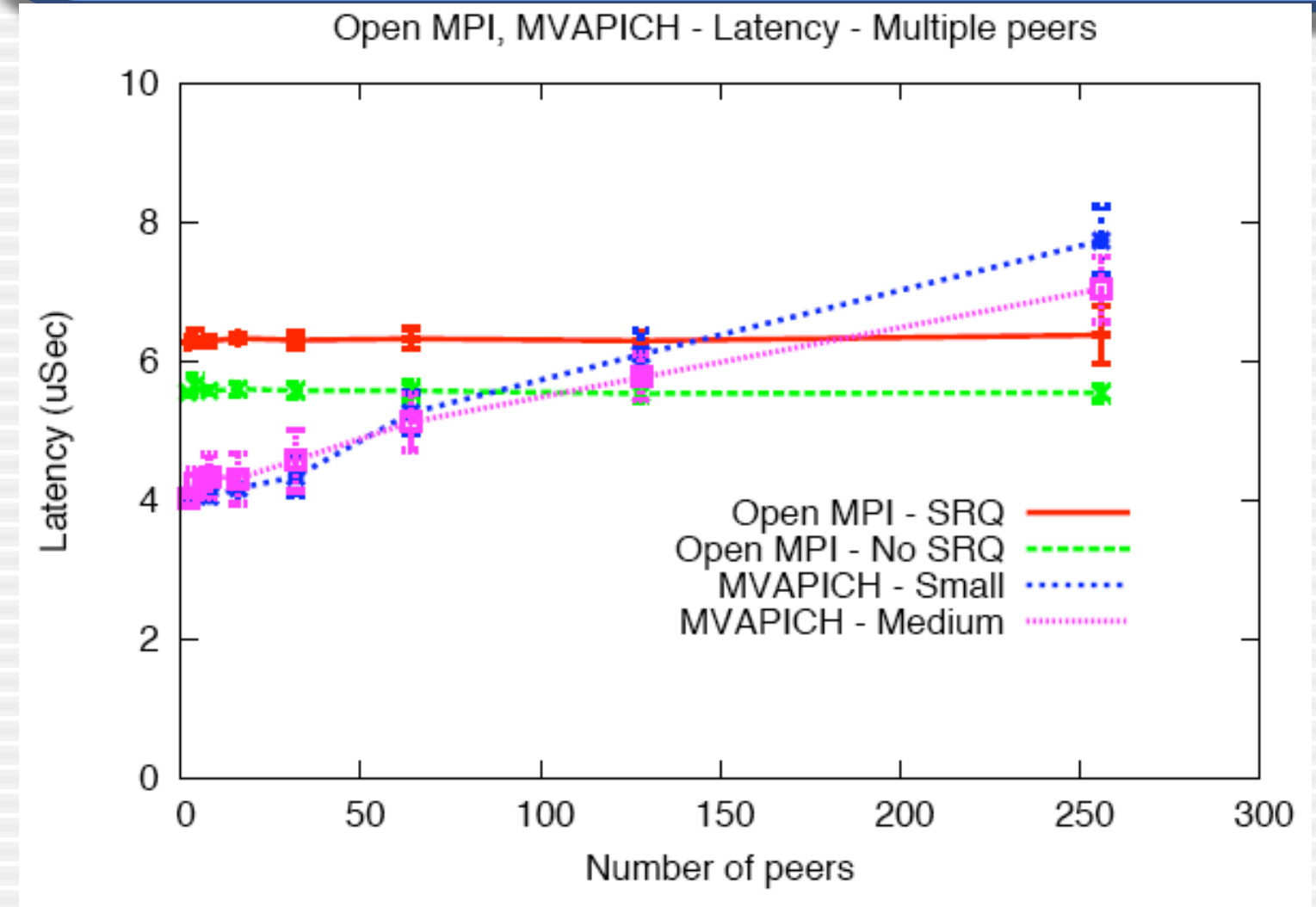
- Current
  - Linux
  - OS X (BSD)
- Not frequently tested
  - Solaris
  - AIX
- Development
  - MS Window
- Maybe?
  - HP/UX, IRIX

- Majority of OMPI is POSIX C
  - Not difficult to port to new OS's
- Segregate OS-specific functionality
  - Plugins

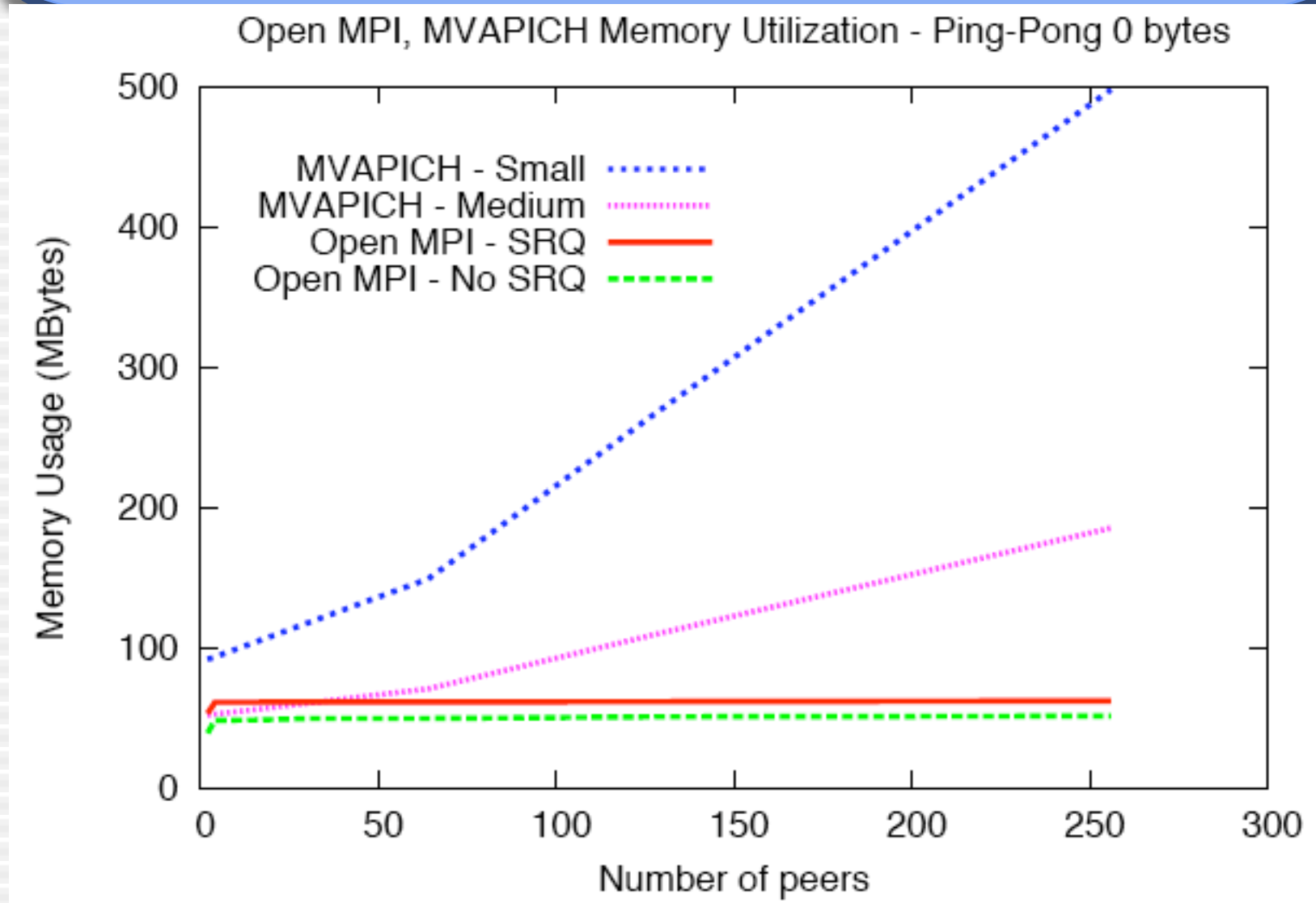
Network Stacks Supported

- Natively support commodity networks
  - TCP
  - Shared memory
  - Myrinet
    - GM, MX
  - Infiniband
    - mVAPI, OpenIB
  - Portals

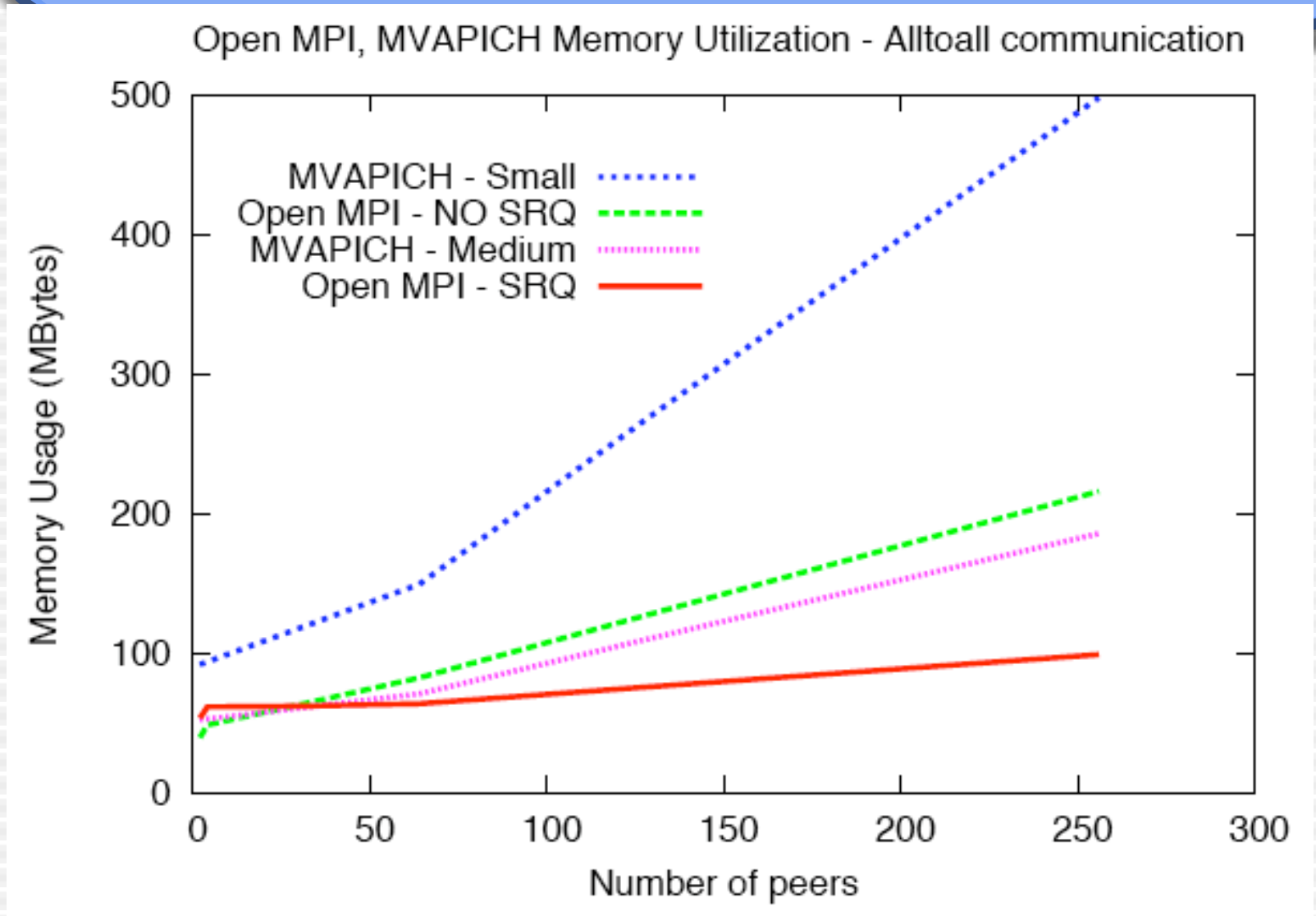
Scalability of Open MPI latencies



Memory Usage

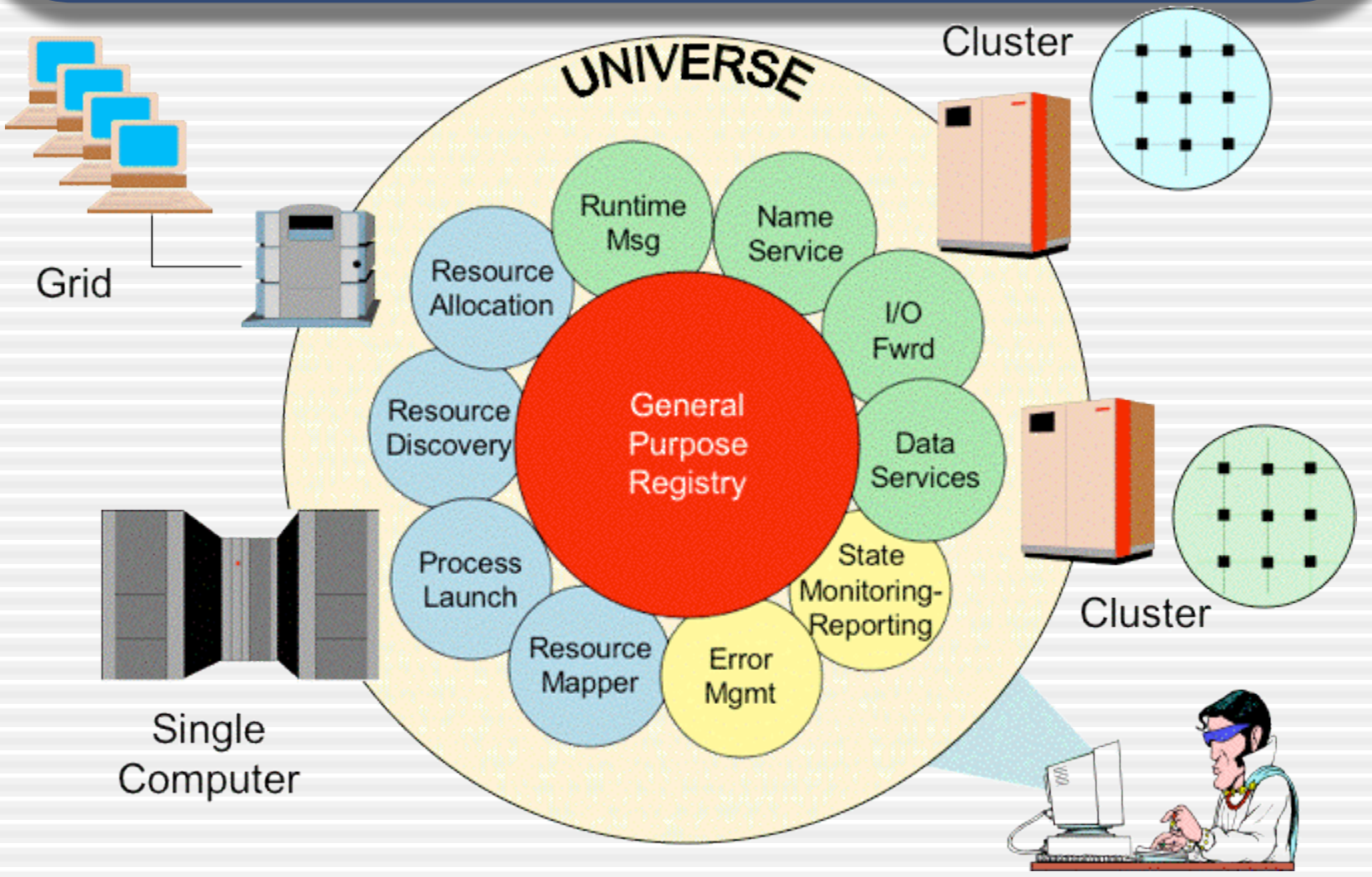


Memory Usage



Open Run Time Environment (ORTE)

OpenRTE Architecture



Run-Time Environments

- Daemon and daemon-less modes
  - vs. LAM/MPI
- Current support
  - rsh / ssh
  - BProc (current)
  - PBS / Torque
  - SLURM
  - BJS (LANL BProc Clustermatic)
  - Yod (Red Storm)
- Future
  - SGE
  - LSF
  - BProc (Scyld)
  - RMS (Quadrics)
  - Grid (“multi-cell”)
- Segregate RTE-specific functionality
  - Plugins

Application Specific Architectures

ASpA Goals

- study, define, build, assess high performance computing architectures that incorporate specialized hardware co-processors
- build software tools to transition scientific codes so that they can effectively exploit heterogeneous computing with co-processors

ASpA Achievements

- R&D 100 award for Trident C-to-FPGA compiler for scientific computing
- Characterized Clearspeed SIMD accelerators
  - 5X on neural net
  - poor performance on monte carlo radiative heat transfer sim, sparse mat-vec

ASpA Achievements - Cont'd

- World class research in Reconfigurable Computing, in collaboration with ISR-3
  - "We wrote the book"
    - Reconfigurable Computing with FPGAs by Gokhale & Graham
  - Two generations of C-to-hardware compiler
  - Reconfigurable Supercomputing Applications
    - Monte Carlo Radiative Heat Transfer Simulation (10X speedup over microprocessor)
    - Metropolitan Road Traffic Simulation (34X speedup over microprocessor)
    - Challenge is Amdahl's law
      - increase the amount of code that can run on the acceleration engine

On-going research

- Characterize microprocessor/FPGA hybrid architectures
- Quantify performance of FPA chips on floating point intensive codes
- Investigate new system architectures
  - ♣ dual socket microprocessor/co-processor
- Study performance impact of FPGA/FPA on large applications
  - ♣ image/video processing
  - ♣ homeland security
  - ♣ numerical codes

On-going research -Cont'd

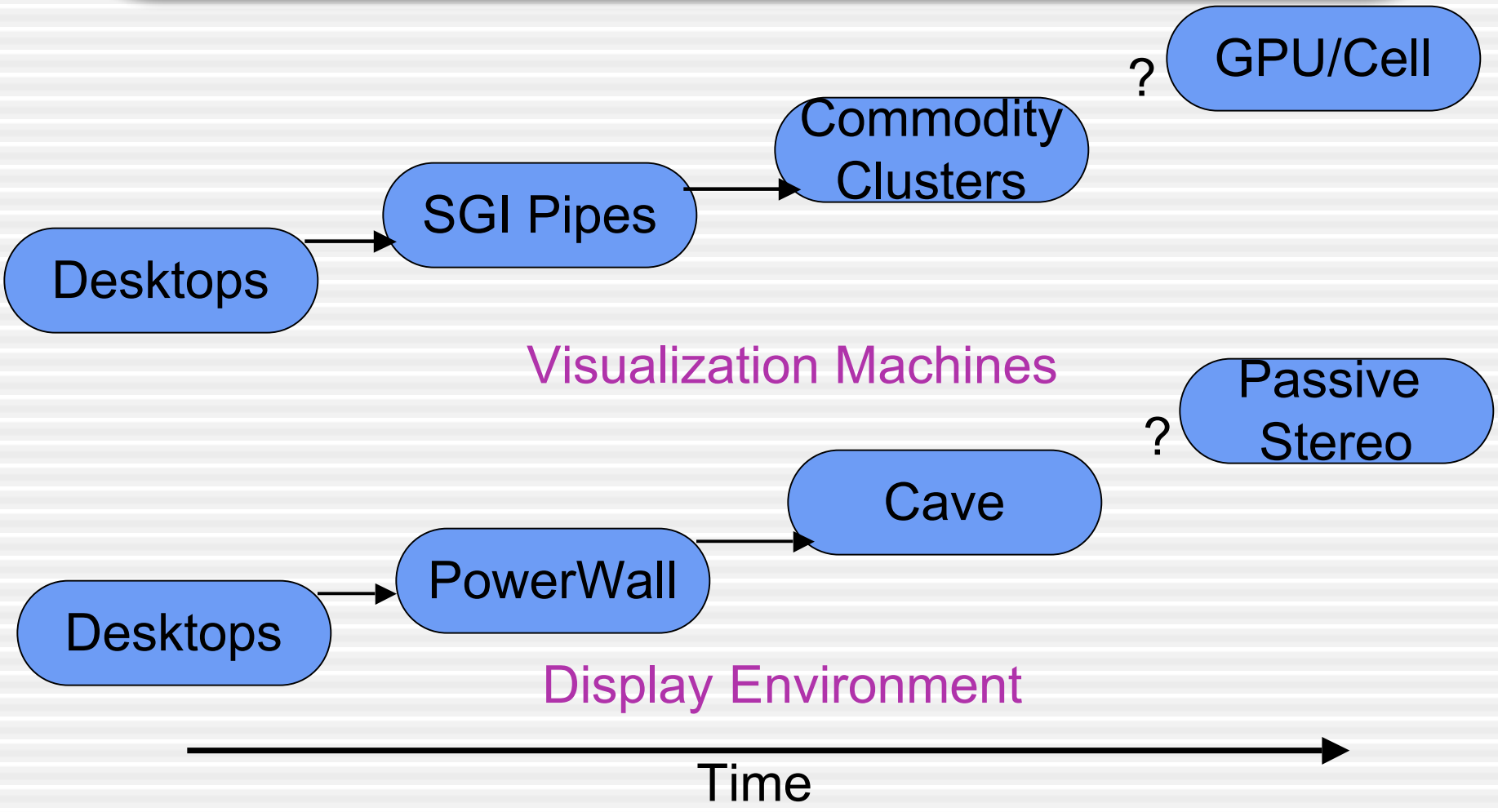
- Develop software tools to compile scientific codes to co-processor architectures
- Nanocomputing architectures
  - ♣ reliability/redundancy tradeoffis in nano-scale programmable fabrics
  - ♣ nano-scale programmable fabric architectures

Visualization

Team Focus

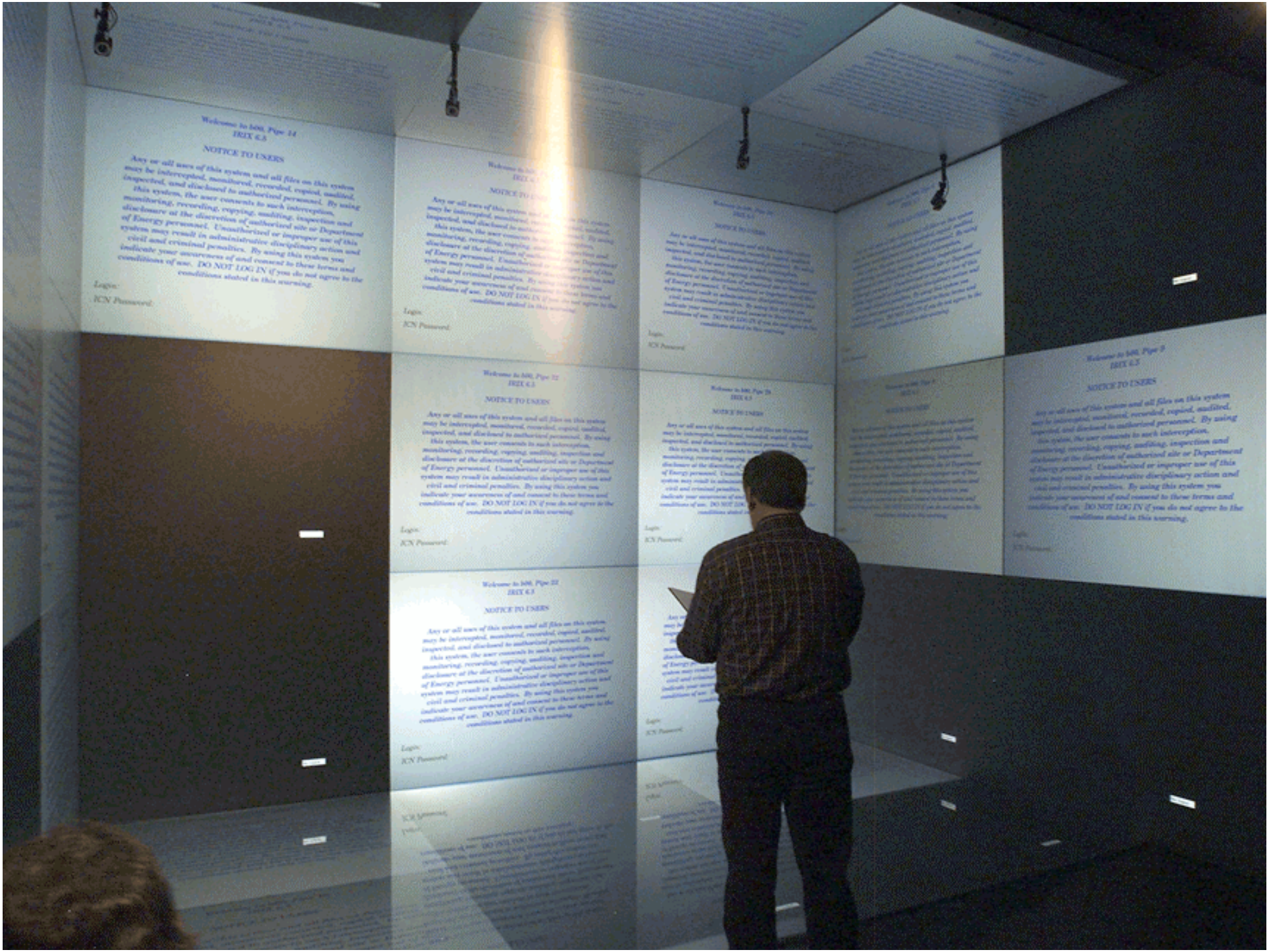
- **Interactive** visualization and analysis of **extremely large**, **time varying**, datasets
- Visualization of such datasets over **large geographic distance**, yet still enabling some interactivity
  - ♣ Each ASC platform is a **remote resource** for 2 of 3 labs!
- Comparative and quantitative visualization and analysis

Historical Perspective




SAGE Cs45e
50.00 sec


/04


Welcome to h00, Pipe 14
IRIX 6.5
NOTICE TO USERS
Any or all uses of this system and all files on this system may be intercepted, monitored, recorded, copied, audited, inspected, and disclosed to authorized personnel. By using this system, the user consents to such interception, monitoring, recording, copying, auditing, inspection and disclosure at the discretion of authorized site or Department of Energy personnel. Unauthorized or improper use of this system may result in administrative disciplinary action and civil and criminal penalties. By using this system you indicate your awareness of and consent to these terms and conditions of use. DO NOT LOG IN if you do not agree to the conditions stated in this warning.
Login:
ICN Password:
Welcome to h00, Pipe 22
IRIX 6.5
NOTICE TO USERS
Any or all uses of this system and all files on this system may be intercepted, monitored, recorded, copied, audited, inspected, and disclosed to authorized personnel. By using this system, the user consents to such interception, monitoring, recording, copying, auditing, inspection and disclosure at the discretion of authorized site or Department of Energy personnel. Unauthorized or improper use of this system may result in administrative disciplinary action and civil and criminal penalties. By using this system you indicate your awareness of and consent to these terms and conditions of use. DO NOT LOG IN if you do not agree to the conditions stated in this warning.
Login:
ICN Password:
Welcome to h00, Pipe 9
IRIX 6.5
NOTICE TO USERS
Any or all uses of this system and all files on this system may be intercepted, monitored, recorded, copied, audited, inspected, and disclosed to authorized personnel. By using this system, the user consents to such interception, monitoring, recording, copying, auditing, inspection and disclosure at the discretion of authorized site or Department of Energy personnel. Unauthorized or improper use of this system may result in administrative disciplinary action and civil and criminal penalties. By using this system you indicate your awareness of and consent to these terms and conditions of use. DO NOT LOG IN if you do not agree to the conditions stated in this warning.
Login:
ICN Password:

Vtk and ParaView - ASC Visualization Research Platform

- VTK
  - An open-source object-oriented visualization toolkit
  - www.vtk.org
- ParaView
  - An open-source, scalable multi-platform visualization application
  - Creates an open, flexible, and intuitive user interface for VTK
  - Project Lead: James Ahrens
  - www.paraview.org

- Past agency funding
  - NSF, NIH, DOE, DOD
- Entities using/developing
  - Laboratories
    - ANL, NCSA, EVL
    - LANL, LLNL, SNL
    - CEA, CHCH
    - ARL
  - Commercial Companies
    - GE, DuPont
  - Universities
    - Stanford, UNC, Utah
- ~2000 mailing list participants

ParaView Overview

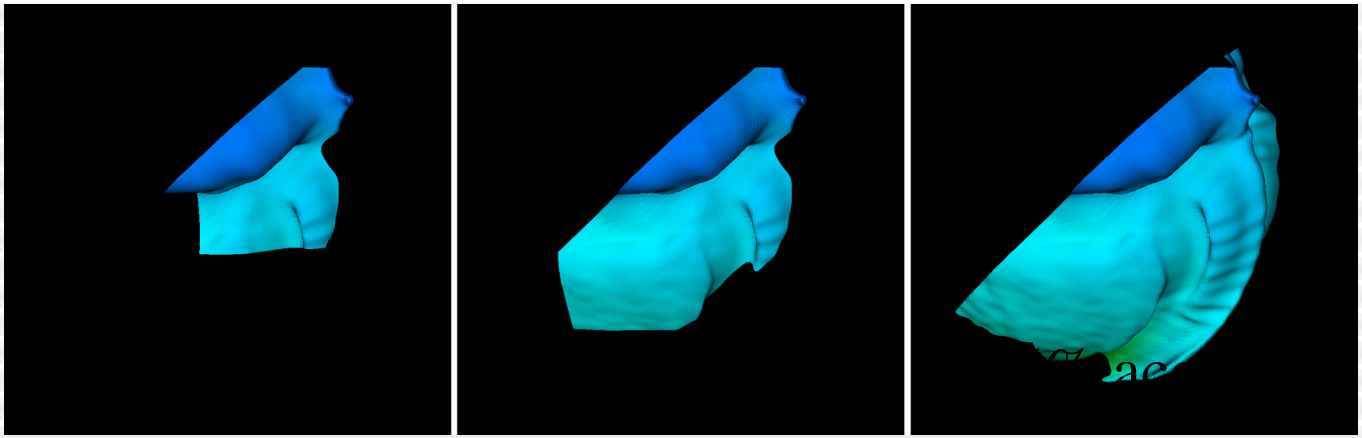
- Serial and parallel portability
  - Run on most serial and parallel platforms
    - Binaries for Windows, Linux, Mac
  - Distributed-memory execution
    - Commodity clusters
  - Comparable performance to PixelVision - no specialized h/w
- Scalability
  - Data parallelism and incremental processing
  - Visualized a petabyte-sized tested problem in 2001
- Advanced displays and rendering
  - Stereo, Tiled walls, CAVE
  - Automatic level of detail rendering
  - Compression for remote data transfer

- Remote visualization services
  - Parallel data server
  - Parallel rendering server
  - Client
- Advance application control
  - Tracing
  - Scripting
  - Animation Editor

Distance Visualization

- Out-of-core
  - Data larger than main memory
- Streaming
  - Incremental processing
- Data divided into pieces and streamed through the visualization pipeline
  - Culling and prioritization
    - Based on value and location
  - Vtk-based – all programs!

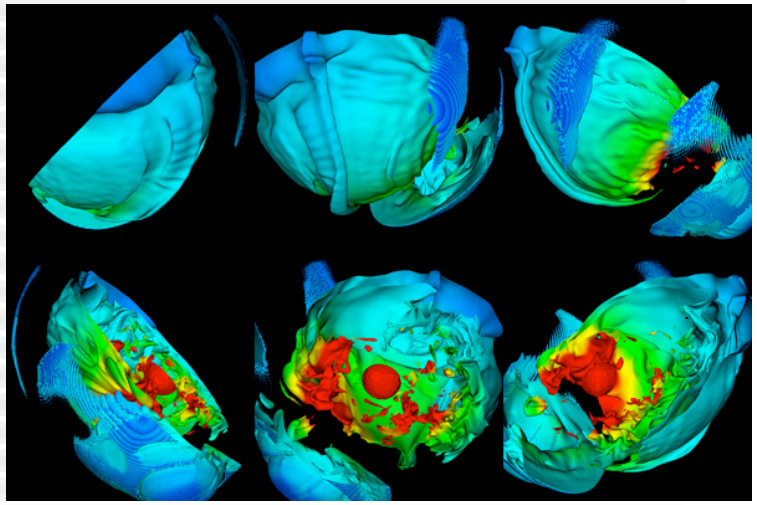
Prioritization

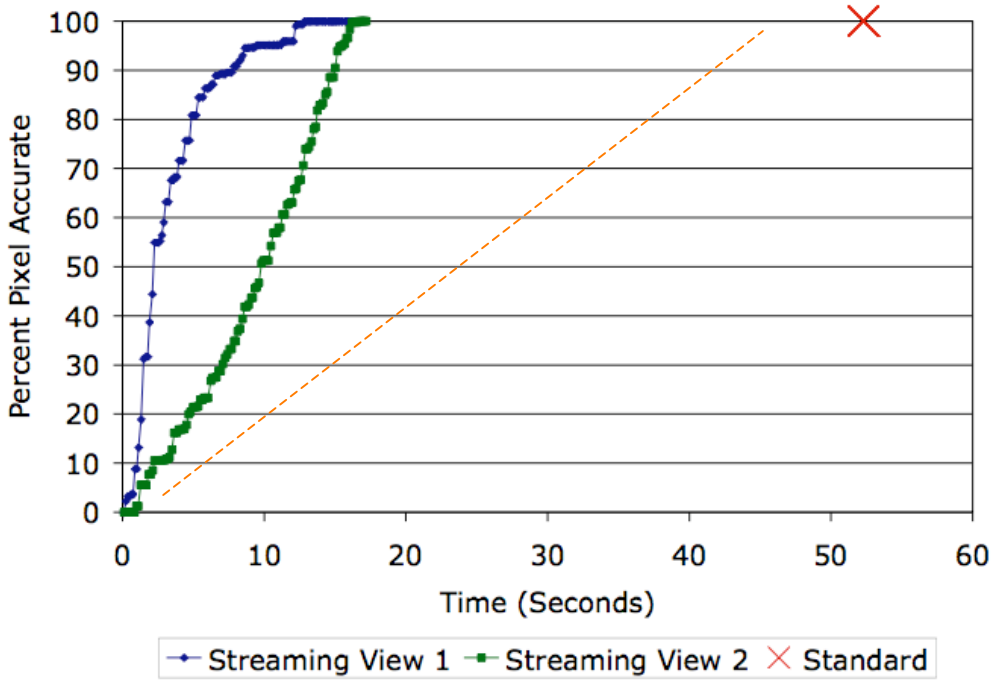


25% 50% 75%

- Results displayed progressively
  - Finished in 2.4%, 3.8% and 7.7% of the time it takes the standard architecture to generate the final image

Image Accuracy Versus Time

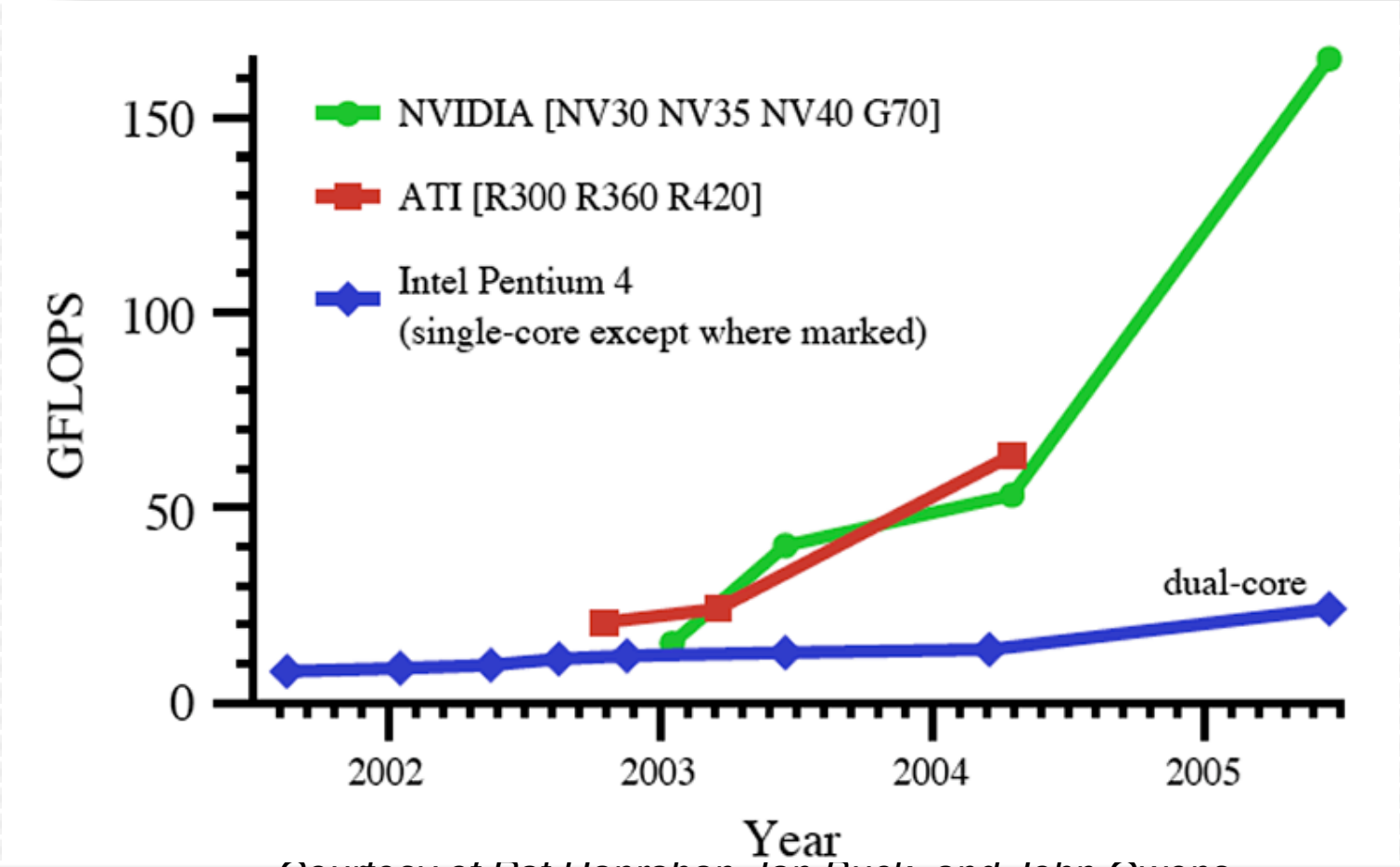




Scout Overview

- High-performance
  - Hardware-acceleration via the GPU
- Quantitative
  - Define and analyze data via programming language
- Scientist-focused programming language
  - Express both general computations and visualization results (GP-GPU)
  - Explicit data parallelism
    - Take advantage of data parallel nature of graphics hardware
  - Hide other nuances introduced by graphics API and hardware

Scout: Hardware-acceleration on the GPU



Courtesy of Pat Hanrahan, Ian Buck, and John Owens.

Future Viz and Petaflop Platforms

- What will a Petaflop platform look like?
- It will not be a cluster of PCs
- Will likely require “unique” architecture
  - CPUs with exotic instruction sets
  - Fast, specialized memory subsystem
  - Fast, specialized processor interconnects
  - May require specialized programming models (e.g. streams)
- Start looking at how these architectures can be used to accomplish interactive and integrated visualization and analysis

IBM CELL Project

- First architecture to evaluate (started in end of ‘04)
- Basis of Sony’s Playstation 3
- Who knows, may be used as basis of Petaflop machine some day
- We are working with Utah and Stanford to evaluate this chip
  - ♣ Volumetric ray casting
  - ♣ Programming language and system issues
  - ♣ Probably on oct-tree based data format
  - ♣ See how fast we can go (interactive?)